Reviews

Comment by the Editor:

The reviewers find the approach presented here interesting, but criticize that you have not established the specific advantage of the presented approaches over existing approaches. We feel it is important to analyse common use cases where existing approaches fail or cannot be applied. So far the only comment on the advantage of SLiM over the other methods seems to be that SLiM can take circular genomes into account (How much does this matter?).

Furthermore, we find it difficult to interpret the data and figures presented in the results section. For example, the data presented in Figure 2: 1. There is no 1 to 1 comparison between the WF expectation and the simulation results. For example, a simulation without recombination would be useful to show that in ideal circumstances the simulations perform as expected. 2. Why/how can the normalization lead to negative values? A better explanation of how the normalization works would be helpful interpreting the figure.

It is also unclear what exactly the figures are intended to show. If the main aim of the figure is to show that rescaling does not have an effect on the data, then the figure should show a direct comparison between different scaling factors. Once it is established that the scaling factors do not change the results, SLiM could then be compared to existing methods. In general, as has been pointed out by the reviewers, improved figure legends would help with understanding the presented data. Finally, jargon and abbreviations are used to an extent that the paper becomes difficult to read.

In conclusion the manuscript requires very substantial revision in order to be recommended. Importantly, we feel a revision should include data regarding the advantage of SLiM over existing methods.

We thank the editor for the comments made. We agree with the comments about the figures, and made the suggested changes. However, we believe there was a misunderstanding about the method we showcase here, as explained to the reviewers below. The adaptation of SLiM presented here is not aimed at competing with FastSimBac or ms on “simple” scenarios but rather to open new simulation possibilities. Still, we compare a simple model of bacterial simulation with other simulators, in order to test whether the newly implemented simulator behaves properly. It appears that it was not clear enough that SLiM is able to simulate a very wide range of scenarios that are out of reach of the other simulators. To clarify this, we improve the text to highlight our goal, and add a new model illustrating the flexibility of the simulator. In this new example, we simulate bacteria growing on a Petri dish with antibiotics in half of the plate and developing resistance thanks to a beneficial mutation. This more complex model gives a good insight into the variety of scenarios that one could simulate.

In general we believe that the manuscript has benefited greatly from addressing the reviewer’s comments and we thank you and the reviewers for your consideration. The major changes are highlighted in the revised manuscript.

Reviewed by anonymous reviewer, 2020-11-06 00:05

Reviewing the paper: Simulations of bacteria populations with SliM

Dear authors, I appreciate your effort in writing this manuscript. Overall I found the manuscript interesting.

Thank you.

Introduction |- Comments

I found the title of the paper a bit deceiving. From the title, I was expecting to see an example of bacterial populations under complex demographic scenarios and selection forces. This paper is more technical. In the first two paragraphs, you explain why simulations are so crucial in bacterial population genomics. Simulation can reveal the past and forecast the new demographic and evolutionary changes of bacterial populations. In your paper, though you do not show any direct evidence of SLiM doing that. You do not show any example where SLiM quantifies the eco-evo dynamics of bacterial populations. In the third paragraph, you compared SLiM to other simulators (e.g., a forward genetic simulator that can simulate complex scenarios including demographics and selection forces, has its language Eidos which makes easily adjustable to simulate bacterial populations).

Answer: This paper is indeed a technical one. We stress the importance of "simple" simulations to verify the correctness of a newly implemented model. A simple scenario also enables a pedagogical walk-through of the different blocks necessary for simulating bacteria with SLiM, so users could easily build more complex scenario from this one. However, we understand that the introduction may have not been clear enough, thus we clarified the aim of this paper, which is bacterial simulation. Notably, we clarified that inferring demography is not doable with SLiM alone, as it requires other methods (such as ABC) to perform the inference of the parameters. We also now showcase a more complex scenario consisting in bacteria and antibiotic interaction on a petri dish to illustrate the potential of our bacterial model.

Action:
- Rephrase part of the intro
- Add example of how to build a more complex model.

Methods |-Comments The methods section confused me so much. For this, I'll go step by step. SLim comes together with the following characteristics: 1. Forward simulator 2. It has its own coding language, Eidos, which makes it adjustable for simulating bacterial populations 3. It allows you to simulate bacterial simulation under the assumptions of a Wright-Fisher model and a non-WF framework. This is quite clear to me. Comment: In this manuscript, you are performing simulations of bacterial populations under the non-WF framework, but you do not validate the non-WF results with any experimental data.

Answer: We understand that comparing the SFS and LD of real bacteria under on constant size scenario with the same parameters as in the model would be helpful, however such dataset does not exist. There is also no experimental data with known demographic and adaptive parameters, and one goal of this simulator is actually to help inferring these processes for real populations.

Methods |-Horizontal gene transfer, recombination and circularity - Comments

Horizontal gene transfer: The exchange of pieces of DNA between different organisms. The piece can be inserted at a random site or a specific site. If the incoming fragment is homologous, then the piece can be incorporated in a way that is similar to gene conversion to eukaryotes, where you do not have a reciprocal exchange of genetic material.

Comment: I see the importance of taking into consideration gene conversion, but you can potentially cite a paper reflecting its importance in the adaptation of bacterial populations, together with the frequency of gene conversion and homologous recombination. Also, you talk so much about gene conversion which at the end you do not consider it in your results, except if you refer to recombination as gene conversion which I doubt. This isn't very clear. You rightly claim that SLiM is superior to other programs because it can simulate gene conversion and because you consider the bacterial chromosome is circular. Why is this important? It is known that a bacterial chromosome, in any case, looks like a smear, a chaotic construction where DNA helixes are entangled with each other. Also, later in the paper, you counter-attack your argument of gene conversion by writing. Because we simulate the entire population; it is not possible to use gene conversion at a significant rate, otherwise ms crashes, thus there is no recombination in "burn-in"

Answer: Bacterial recombination occurs through a process similar to gene conversion in eukaryote and not by cross-over. When referring to recombination throughout the text we are thus referring to bacterial recombination. We rephrase the text to make it clearer.

Models including circularity have been studied in the past and it was shown that circularity can lead to different patterns, such as LD decaying faster in linear genomes than in circular genomes (Wiuf 2001, Robinson et al book). Although circularity is likely not important for the metrics and parameters shown in this study, including it is one less incorrect assumption when modeling bacteria. We understand that emphasizing too much this feature might be misleading and thus we toned down this aspect. After modifications we hope we clarified that the advantage of SLiM compared to other programs is that it can indeed simulate bacterial recombination (with a process similar to gene conversion) together with a wide variety of scenarios.

Action: Improve section on Horizontal gene transfer, recombination, and circularity.

Methods |-Burn-in - Comments

It is desirable to start a simulation with a population that is in a mutation-drift equilibrium. We have a mutation-drift equilibrium when both the mutation rate and

the effective population size are stable. In a mutation drift equilibrium, the rate that the variation is lost due to drift is the same that is gained due to mutation.

Comments: I do not understand what does it mean when you say that the population size is larger than the time-span of interest guess you mean the effective population size that is needed to reach a mutation drift equilibrium is very high. Could you clear this out?

Answer: This section was indeed not clear enough, and we rephrase it to improve the understanding of the point we make. What we meant was that doing a burn-in with a forward simulation is not practical because it takes too much time. Indeed the effective population size of bacteria are often much larger than the period of interest. In our case, the effective population size is 140,000, while the period of interest is 20,000 generations (much smaller than 140k). In this situation, a burn-in with a forward simulator would last 5x140k = 700 thousand generations (5.Ne). So the period of interest would represent here about 2% of the total simulation. Besides, 5xNe generations does not offer a guarantee that we reach this equilibrium.

Action:
- We rephrase the section
- We added details about this expectation in the Supplementary Materials.

Methods |- Simulation rescaling - Comments Here you discuss the effect of rescaling into the summary statistics of the program. It's quite clear to me.

ok.

Methods |- Simulation protocol - Comments

Overall the simulation protocol is detailed and well explained. Many times, however, I was getting errors when I tried to copy-paste the code in the SLiMgui (e.g., ERROR (EidosSymbolTable::_GetValue): undefined identifier genomeSize. This error has invalidated the simulation; it cannot be run further. Once the script is fixed, you can recycle the simulation and try again) I suggest making the code more accessible, so when we test the code of the paper not to paste the line numbers as well. However, I see the importance of enumeration. In the end, I used your GitHub code where enumeration is hard to be followed.

Answer: We did not anticipate that readers would use the snippet with SLiM GUI. However we found that it is a great idea and we thus modify the code snippet to make this experience possible.

Action: We declare the variable within the SLiM script, and not from the bash command line. We tried to make the line number not selectable anymore when copying and pasting, but it appears that it depends on the pdf reader.

Results The Results are quite straight forward. However, when I was reading your introduction, I was prepared for a different type of results. You did what you wrote about at the end of the introduction (you introduced the model, and that model behaves according to WF-model). Still, you also present a non-WF model whose results you do not validate from experimental data.

Answer: See answer above regarding the absence of appropriate experimental data.

Figure1: rescaling ~ CPU time and memory Figure2: SFS ~ rescaling Figure3: LD ~ rescaling Figure 4: recombination rater & tract length ~ CPU & memory Figure 5: SFS ~ recombination rate Figure 6:

Comment: With the caption of your figures, you should convey the main result of the figure to be easier for the reader to skim through your soon to be published. For example, in Figure 1, you could write that by increasing the rescaling factor you observer faster CPU time, and less memory and that nonWF pops are being faster.

Answer: We agree with the reviewer that the caption were not conveying the important results..

Action: We improve the legends of the figures

Discussion

In the discussion, you summarise your results and refer to the drawbacks of your simulator. I could not even find a typo. In general, I have to admit that I admire your efforts. The paper is neat, well structured, even the bibliography is written accurately. However, there is a space for improvement. Your methods section I believe that needs to be written more clearly. There are several points where the reader gets confused. You have to make from the introduction very clear your points, do not refer to gene conversion as your strong point since it is not, clear out what do you mean by recombination, pass out that this is technical paper.

Answer: We thank the reviewer for the remarks and comments made. We hope that the modifications made now render the paper clearer both in term of its goal and of its methodology.

Reviewed by anonymous reviewer, 2020-10-28 15:59

Simulating genetic data under various scenarios is standard practice in evolutionary biology. This is usually done using coalescent simulations, which work well for neutral models. The stated aim of the paper under review is to “go beyond the limitations of the coalescent” (p. 2).

The authors pursue this by showing how the forward simulator SLiM, apparently first published in 2013, can be adapted to bacterial populations; its original target are eukaryotes. The results are similar to those obtained with classical tools such as ms. And while ms is much faster than SLiM under most scenarios, it is overtaken under high recombination. The paper is essentially an addition to the SLiM manual, it contains no new biology or algorithm. However, SLiM appears to be a good forward simulator and there are bound to be scenarios not covered by current coalescent simulators, though fast gene conversion isn’t one of them. The authors state that ms gets slow with high recombination and msprime so far lacks gene conversion. However, macs is a practical simulator with fast gene conversion published in 2008, which isn’t mentioned.

Answer: We thank the reviewer for his/her remarks. We indeed reuse SLiM to make usable for bacteria, by notably implementing bacterial recombination (i.e. recombination between any two individuals in a gene-conversion fashion). The goal of the comparison with ms and fastSimbac was only to compare the resulting summary statistics because theoretical expectations are hard to derive, especially for LD. The aim of this paper is not to show that our simulator is faster than ms with high recombination, but rather to show that it behaves similarly (in terms of generated populations) and in a reasonable amount of time. This simple comparison helps verifying that our model is correct. SLiM's main advantage is its capacity to simulate a very wide range of possible scenario, which coalescent and markovian coalescent simulators cannot perform.

Action: We rephrased when appropriate to clarify that our main goal is to showcase a simulator that is more flexible and yet efficient.

Here are a few additional detailed comments:

1. p. 2: The authors criticize that gene conversion in ms is based on a linear chromosome, whereas SLiM implements the circular chromosomes found in bacteria. Where does this make a difference?

Answer: We replied above in reviewer 1: Briefly, The circularity is likely not important in the metrics we show, however it is one assumption less when modeling bacteria. However we understand that we may have oversold this feature and thus we tone down this aspect

2. p. 3: The authors recommend a burn-in of 5Ne generations, but caution that this also does not guarantee equilibrium. What is the probability of reaching equilibrium as a function of burn-in length? Or is that not known?

Answer: The reviewer is right that 5Ne generations does not guarantee equilibrium.

Action: We rephrased this section to make it clearer to the reader, and we added an appendix to justify the 5.Ne rule of thumb.

3. p. 7: All entries in Figure 1 should be based on the same number of replicates, even if times need to be extrapolated from smaller runs.

Answer: Because we are not interested in comparing extreme values (outliers) of these distributions but merely the median and quartiles, differences in the number of replicates (sampling sizes of each box) are not problematic (Krzywinski & Altman 2014). In addition the smaller number of replicates is 30 which is large enough for showing boxplots (the minimal value is 5). When comparing these quantities across different sample sizes only the precision might vary, but not the expectation. On the other hand, extrapolating running times and memory usages from shorter runs could introduce unexpected biases. We followed the common statistical analyses recommendations by presenting the numbers of replicates for each experiment in the main text, section 3.1 and are now including these numbers in the figure caption.

Krzywinski, M., & Altman, N. (2014). Visualizing samples with box plots: use box plots to illustrate the spread and differences of samples. Nature Methods, 11(2), 119-121.

4. p. 7: What is the memory requirement of ms and FastSimBac in the lower panel of Figure 1?

Answer: We added the average memory requirement for ms and FastSimBac runs in Figure 1.

5. p. 10: Recombination makes ms slow, but a more appropriate comparison might be to macs.

Answer: As explained above, our goal is not to have the best simulator in terms of how well it handles high recombination rate, but to have a flexible (thus forward in time) and efficient simulator.

6. p. 10: As in Figure 1, the run times should be based on the same number of replicates.

Answer: See answer from above.

Reviewed by anonymous reviewer, 2020-10-22 06:26

This manuscript describes how to adapt the popular simulator SLiM to bacteria, especially to the bacterial mode of recombination. I had wondered about this possibility myself in the past, and I am delighted to see this preprint and the described protocol. However, I see several possibilities for improving the manuscript to better highlight the improvements of the described approach compared to existing approaches.

We thank the reviewer to stress the importance of our work.

The manuscript would greatly profit from an overview figure that explains the underlying model and the different parameters used and how they go into the simulation.

Action: We added a simple schema representing the model and the different parameters.

The main advantage of the described approach should be presented with an example and discussed. So far, only simulations with comparisons to other programs are done, and they show convincingly that the SLiM approach works well. However, it is not obvious which advantages the presented approach has compared to ms and FastSimBac. Maybe one more complex simulation that includes selection or population structure could be added in the end to show an application of the approach. The advantages over previous approaches could also be added to the "Discussion" section.

Answer: This is a remark that other reviewers have made, thus we improved the clarity of the text and added a more complex model which, we hope, will give readers an idea of the breadth of possible scenarios.

Action: We added a very different scenario: Growth of 50 colonies on a petri dish, half of which is covered by an antibiotic, and visualisation at different time steps of the growth and of the resistant bacteria colonies (which has a cost without antibiotic). We believe such model illustrates the wide range of possible scenarios.

Section 2.2.1 "mean recombination tract length of 10kb" First, the distribution could be mentioned here already, although this can be seen in the code. My main point is, however, that this value appears quite large. E.g., unselected recombination events found in 10.1371/journal.ppat.1002745 are on average 2kbp, most of the recombinations inferred in 10.1128/mBio.02494-18 are below 10kb, and the average length of homologous recombination fragments inferred in E. coli is ~500bp (10.1186/1471-2164-13-256). The simulation is presented for parameters from S. agalactiae where the mean length is even above 100kb, and this paper is based on selected recombination events, whereas unselected events should provide the parameters for the simulation (see 10.1371/journal.ppat.1002745 for the difference). Although, I understand that these parameters can be adjusted, I wondered how the simulations perform for shorter length.

Answer: In section 3.2, "impact of recombination", we perform simulations with different values of the mean recombination tract length. In this experiment the parameter varies from lambda/100 to lambda, i.e. from 122kb/100 = 1220 bp to 122kb. For a short length of 1220bp, we can confirm that the model behaved correctly (Figure 5 and 6).

Action: We added the mention that it is from a geometric distribution in the text, and not only in the code.

Section 2.2.1 It is not clear to me how the source individual for the recombination event is chosen. Since offspring is directly added to the population, is it possible, that generated recombinants can already be the source individual for recombinants generated later in the same generation?

Answer: No it is not possible, newly generated individuals are held off to the side until reproduction() callbacks finish executing, and are then merged into their respective subpopulations, specifically to prevent this type of behaviour.

Action: We precise this point.

The authors should mention the recently released simulator CoreSimul (10.1186/s12859-020-03619-x), maybe in the introduction. If feasible, it would be interesting to see how it compares to SLiM.

Answer: We thought about the possibility to add CoreSimul in the introduction but as this simulator simulates slightly different data from what we had in mind we chose not to. But reconsidering our first thought, we can definitely mention CoreSimul as a bacterial simulator stressing its difference (mainly simulating nucleotide with model of sequence evolution). We also tried CoreSimul with some

of our parameters and the tree given in example (34 individuals) , but it was too slow, most likely because of the length of the simulated chromosome. For instance, it takes about 100 seconds to simulate a fragment of 200kb, and takes much longer for 2Mb, running in about 7000 seconds (about 2h). We understand that CoreSimul does not simulate populations but rather coregenomes of a sampled population, and thus an accurate comparison is not straightforward. We believe each simulator has its own advantages.

Action: Mention of CoreSimul in the introduction.

Additional comments: Section 2.1.2 "Because we simulate the entire population, it is not possible to use gene conversion at a significant rate, otherwise ms crashes, thus there is no recombination in burn-in." Maybe you can be more precise and describe why ms crashed, would more RAM solve the issue? Which population size would the feasible with ms?

Answer: It was a segmentation fault caused most likely by lack of RAM. For the record, with our set of parameters and a rescaling factor of 100 (thus a population size and sample size of 1400 individuals) and even when dividing the recombination rate by 100, ms requires more than 15Go of RAM. This is already a lot, considering that a realistic recombination rate (100 times higher) would lead to considerably more recombination events and required even more RAM. We are confident that more RAM would not solve the issue, on a practical scale.

Section 2.1.3 "The rescaling factor must also be applied to the duration of the simulation (and the duration of different events that might occur), so that the effects of drift remains similar." Maybe it could be described explicitly how the length and events should be increased or decreased.

Action: We added an example to precise how the rescaling should be applied for the duration of events:

“For instance, with a rescaling factor of 10, the length of the simulation will be shorten by a factor of 10, as well as any other events, like the length of a bottleneck, or the start time of an expansion.”

Section 2.2.1 "constant 11" Should it read "constant 1"?

Answer: It should read “defined by the constant line 11”.

Action: fix typo.